University of Tripoli/faculty of engineering
Electrical & Electronic Eng. Dept.

Course: EE334 FIRST EXAM SPRING 2013

Q-1) a : Describe the operation that 8086 will perform when it execute each of the instructions :

1- MOV BX, 03FFh ; LOAD BX WITH AN IMMADIATE VALUE = 03FFH
2- MOv AL ,0DBh ; LOAD AL WITH 8 BIT NUMBER DBH (IMMADDIATE VALUE)
3- MOv DH ,CL ; COPY THE CONTENT OF CL REGITSER (8 BIT) TO THE DH REGISTER (8BIT)
4- MOV BX ,AX ; COPY THE CONTENT OF AX REGITSER (16 BIT) TO THE BX REGISTER (16BIT)

b: Write the 8086 assembly language statement which will perform the following operation

1 – load the number 7986H into the BP register : MOV BP , 7986H
2- copy BP register contents to the SP register : MOV SP , BP
3- copy the content of AX register to the DS register : MOV DS ,AX
4- load the number F3H into AL register : MOV AL , F3H

C: if the data segment register (DS) contains 4000H , what physical address will instruction MOV AL ,[234BH] read ?

DS =4000H , OFFSET IS 234BH SO THE PHYSICAL ADDRESS IS 4234BH

D : If the code segment for an 8086 program start at address 70400H, what number will be in the CS register?

70400H = 5 digit == PHYSICAL ADDRESS ,, PROGRAM BEGIN WITH THIS ADDRESS SO IP =0000H
WHICH MEANS CS = 7040H

E:Write an ALP to find out decimal addition of sixteen four digit decimal numbers?

```
segment data_seg
          ; msg  db "+$"
ends data_seg
segment code_seg
start:
    ; set segment registers:
   mov ax, data_seg
   mov ds, ax
   mov bx,0h
   mov cx,0h
   mov bp ,0h
   mov sp ,0
   mov si ,0010h
               ; 16 four decimal digit
               ; max value of the sum
              ; will have 5 digit in the
              ; following rigster bp-ch-
              ;cl-bh-bl , where bl LSD
              ;,BP MSD
y:     call read
      mov dl ,00h
     add bl ,al
  CMP bl ,09h
  jna cor1
   add bl , 06h
      and bl , 0fh
      mov dl , 01   ; CF= 1
cor1:    call read
      add BH  , dl
```

```
mov DL  , 00h
 add BH  , al
 CMP BH  , 09h
 ; if the answer   above 9 need ;
;adjust by adding
  jna cor2          ;6 to the answer
   add BH , 06h
and Bh , 0fh  ; to clear the upper 4
                          ;bit
 mov dl , 01h   ; CF= 1
cor2:
  call read
  add cl , dl
  mov DL  , 00h
   add cl , al
   CMP cl , 09h
   jna cor3
   add CL , 06h
   and CL , 0fh
   mov DL , 01h   ; CF= 1
cor3:
 call read
 add ch ,DL
 mov DL  , 00h
 add ch ,al
 CMP ch ,09h
 jna cor4
 add ch ,06h
 and ch ,0fh
 inc BP
```

```
cor4:
  call enter
    dec si
     jnz y
      cmp bp ,09
 jbe z
      add bp,06h
 z:   mov ah ,4ch
       int 21h
       enter proc near
              mov dl ,'+'
              mov ah ,02h
              int 21h
              ret
             enter endp
 read proc near
           mov ah, 1
           int 21h
          sub al,30h
          ret
 read endp

ends code_seg
end start
```

===

Q-2) a: How many address line does an 8086 have ?

20 ADDRESS LINE (A0 –A19)

b: how many memory addresses does this number of address lines allow 8086 to access directly ?

$2^N = 2^{20}$ ADDRESS 'N =NUMBER OF ADDRESS LINE'

C: at any given time , the 8086 works with four segment in this address space. How many byte are contained in each segment?

64KByte ON EACH SEGMENT

D:what is the main differences between the 8086 and 8088 ?

DATA BUS 8088 (8 DATA LINE) 8086 (16 DATA LINE)

pin #28 8086 : M/IO' 8088 : IO/M'

pin #34 8086 BHE'/S7 8088 : SS0

E: Describe the function of the 8086 Queue? And does the queue speed up processing?

The BIU feeds the instruction stream to the execution unit through a instruction queue register.
EU simply reads the next instruction byte(s) form the queue register in BIU .
Queue register speed up the execution time by fetching next instruction while EU execute previous instruction.

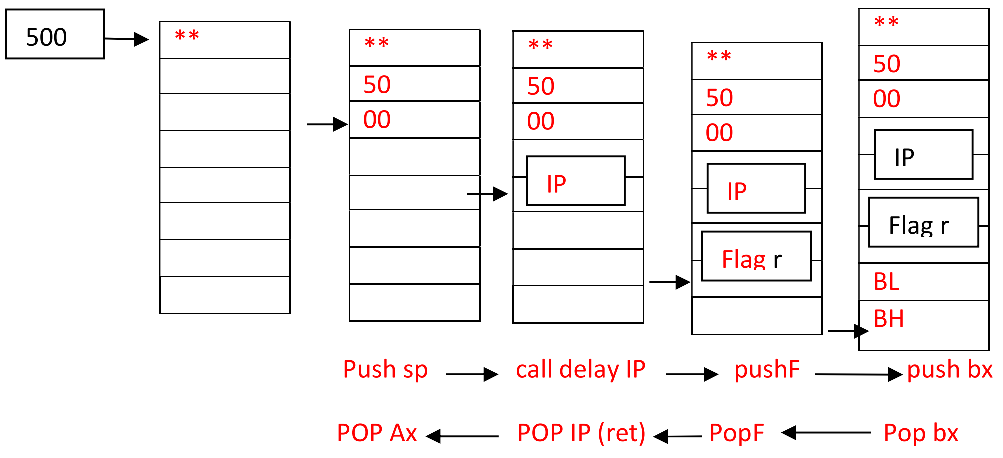
F: Write an ALP to convert a given sixteen bit binary number to its access-3 equivalent?

```
 .data
 x dw xxxxh
.code
  .startup
  mov ax ,x
  mov dx,0
  add  ax, 3
  adc  dx ,0
  ret
 .exit
```

G: Use stack map to show the effect of each of the following instruction on the stack pointer and the content of the stack

```
      MOV SP , 5000H
      PUSH SP
     CALL delay
     POP AX
     ……….
     ………
     ………
    ……………….
```

```
Delay proc near
  PUSHF       ; push a 16bit flag register
  PUSH BX
  .
  .
 pop BX
 popf         ;  POP a 16bit to flag register
 ret
delay endP
```



==

Q-3) a : given the following data structure

```
XXXX  segment
    old      DB    'Libya is free'
     new     DB    13  DUP (0)
XXXX     ends
```

write program which moves the string "Libya is free" from **old** to **new** which just above the initial location?

```
XXXX  segment
    old      DB    'Libya is free'
     new     DB    13  DUP (0)
XXXX     ends
code_seg  segment
     assume Cs:code_seg  , DS:XXXX
     start:
       mov bx ,offset old
      mov   si ,000dh
     z:  mov  al  , [Bx]
        mov   [bx+000dh], al
        inc bx
        dec   si
      jnz  z
     ret
code_seg   ends
end start
```

b: Describe the function of each assembler directive and instruction statement in the program below

```
                    ; pressure read program
        DATA_HERE  SEGMENT
           PRESSURE DB  0                        ; STORAGE FOR PRESSURE
           PRESSURE_PORT          EQU    04H
          CORRECTION _FACTOR  EQU    07H
        DATA_HERE   ENDS
     CODE _HERE  SEGMENT
         ASSUME   CS:CODE_HERE  ,DS:DATA_HERE
               MOV AX , DATA_HERE
              MOV DS ,AX
              IN  AL , PRESSURE_PORT
              ADD AL  CORRECTION _FACTOR
```

```
        MOV PRESSURE  ,AL
   CODE_HERE ENDS
    END
```

==

Q-4) A : write a procedure which produces a delay of 3.33ms when run on 8086 with 5MHz clock ?

3.33msec == 16650 cycle ,, by using NOP and LOOP inst we need 16650/20 rotation

```
mov cx , 832d
x :  nop
   loop x
```

B : write a mainline program which uses this procedure to output a square wave on bit D0 of port FFFAH?

```
.code
```

```
                .startup
               mov dx ,fffAh
             xx:  mov al ,01h
               out  dx, al
               call delay
               mov  al ,00h
              out  dx ,al
              call delay
              jmp xx

Delay  proc  near
     mov cx , 832d
x :  nop
      loop x
     ret
  delay endp
```

=====================================GOOD LUCK ===